

# BOOTING WINDOWS 2000

**After reading this chapter and completing the exercises,  
you will be able to:**

- ◆ Understand the Windows 2000 boot process
- ◆ Troubleshoot system restoration by using Safe Mode
- ◆ Explain the operation of the key Windows 2000 startup files
- ◆ Understand the boot options offered via the Advanced Options menu
- ◆ Edit the Boot.ini file to manipulate the boot process
- ◆ Understand how multiboot configurations are created and how they function

**O**n the surface, booting a computer seems simple. But in reality, booting is a complex process. In fact, it is important to understand each step of the process by which an inert hunk of metal becomes a computer running Windows 2000. This understanding is essential for the Microsoft certification exam, and for troubleshooting a system that won't boot properly.

In this chapter, you learn the steps that Windows 2000 takes to successfully complete a boot process. The process begins with the initial operation of the hardware, as it finds pointers to the software that ultimately leads to the choice of which operating system to run (and as it goes through the process of loading and starting Windows 2000). The process culminates when a successful user logon occurs. It is only at this point that the Windows 2000 boot process is considered to be complete.

---

## THE BOOT PROCESS

All computers, whether hosting Windows 2000 or another operating system, go through a similar **boot process** when they are turned on. In Windows 2000, this process is broken down into two major phases, the **boot phase** and the **load phase**.

The Windows 2000 boot process is actually a two-part process that includes both the boot phase and the load phase. The boot process takes place when the computer is first powered on, and when you choose Shutdown, Restart from the Shut Down Windows dialog box. This dialog box appears when you select Shut Down from the Windows 2000 Security dialog box that takes over your screen any time you enter the Windows 2000 attention sequence, Ctrl+Alt+Delete. It also appears when you select Restart from the Shut Down Windows dialog box (Start, Shut Down).

---

## WINDOWS 2000 BOOT PHASE

The six steps of the Windows 2000 boot phase are as follows:

1. Power-on self test (POST)
2. Initial startup
3. Boot loader
4. Selecting the operating system
5. Detecting hardware
6. Selecting a configuration

### Power-On Self Test (POST)

The **power-on self test (POST)** is the first step in the boot sequence for any computer with an operating system. The POST determines the amount of real memory, and whether or not all necessary hardware components, such as a keyboard, are present. The actual tests can differ, depending on how the BIOS is configured. If the tests are successful, the computer boots itself. If the tests are unsuccessful, the computer reports the error by emitting a series of beeps and possibly displaying an error message and code on the screen. The number of beeps indicates the error, but differs from one BIOS to another. The software for the POST resides in a special, battery-powered chip called the **CMOS (complementary metal-oxide semiconductor)**. This chip can store not only the software necessary to conduct the POST, but also basic configuration information that the POST uses to check the amount of RAM installed in a system, along with other key information. Figure 14-1 shows a typical screen that results from the successful completion of the POST on an Intel PC.



**Figure 14-1** The POST display on a PC

After the system POST is completed, each adapter card in the system performs its own self-test. For example, if a computer has a SCSI card in addition to its own built-in adapter cards, it will check its internal configuration and any related devices it sees when it runs its own POST. At the same time, a report on what it finds during this process will appear on the computer monitor in text-only form (because there is no real operating system running at this point, screen output at this stage of the boot process is kept as simple and direct as possible). The screen shown in Figure 14-2 adds the report from an Adaptec 2940 SCSI controller to the information already supplied by the POST routine.



**Figure 14-2** Output from the BIOS on an Adaptec 2940 SCSI controller

## Initial Startup

The initial startup sequence involves numerous files and initialization procedures. The first sector of the hard disk contains the Master Boot Record (MBR) and the partition table. The

**Master Boot Record (MBR)** begins the boot process by looking up the partition table to determine which partition to use for booting. If you are booting from a floppy disk, the first sector contains the **partition boot sector**.

Table 14-1 outlines the startup files for Windows 2000 on Intel (x86) computers.

**Table 14-1** Windows 2000 Startup Files

Filename	Location	Explanation
Ntldr	Root of startup disk	Windows 2000 boot loader for PC machines
Boot.ini	Root of startup disk	Windows 2000 PC boot menu information
Bootsect.dos	Root of startup disk	Provides DOS boot information for dual-boot PCs
Ntdetect.com	Root of startup disk	Windows 2000 hardware detection program
Ntbootdd.sys	Root of startup disk	Lets Windows 2000 access SCSI drives on PCs with SCSI controller with onboard BIOS disabled
Ntoskrnl.exe	%systemroot%\System32	Windows 2000 operating system kernel
Hal.dll	%systemroot%\System32	Hardware abstraction layer code (CPU driver for x86 chips)
SYSTEM key	%systemroot%\System32	Key Windows 2000 Registry data
Device drivers	%systemroot%\System32	PC-specific device drivers for Windows 2000 use

When the POST has successfully concluded, the BIOS tries to locate the startup disk. The **BIOS** (basic input/output system) represents a chip-based set of routines that DOS and Windows 95/98 use to drive all system input and output, including access to peripheral devices of all kinds. Windows 2000, on the other hand, uses its own built-in input/output logic and drivers, and ignores whatever BIOS is installed in a computer. By doing this, Windows 2000 is able to manage I/O much more carefully than earlier Windows and DOS operating systems. It also helps to explain why applications that attempt to access drivers or the computer's BIOS or hardware directly are treated as ill behaved in the Windows 2000 environment.

If a floppy disk is in drive A when the BIOS checks that drive, it may decide to use that drive as the startup disk (this decision depends on how the boot sequence has been configured in the PC's CMOS). If there is no floppy disk in that drive, or if the CMOS has been configured to boot from a hard disk, it will use the first hard disk it finds as the boot disk. Of course, if drive A is enabled for booting, and the floppy disk you have inserted in that drive does not have a partition boot sector, you will get a “Non-system disk or disk error: Replace and press any key when ready” message, and the system won’t start. This is one of the most common causes of boot failure in the Windows 2000 environment.



If you get the “Non-system disk or disk error” message because the system attempted to boot from a non-system floppy, remove the floppy and cycle the power off and on again. It is important to do this (rather than restarting with Ctrl+Alt+Delete) to avoid transferring boot-sector viruses to the computer.

When the BIOS uses the hard disk as its startup disk, it reads the MBR and loads that into memory. The BIOS then transfers system control to the MBR. The MBR scans the partition table to locate the system partition. When the MBR locates the system partition, it loads sector 0 of the partition into memory, and executes it. Sector 0 can contain a diagnostic program, a utility such as a virus scanner, or a partition boot sector that contains the startup code for the operating system. Should the computer boot from a floppy, only the partition boot sector is used.



In general, the MBR is independent of the operating system. For example, the same MBR is used in x86 systems to boot to Windows 95, MS-DOS, Windows NT, Windows 2000, and Windows 3.x.

However, the partition boot sector is completely dependent on the operating system and file system in use. For example, the partition boot sector in a Windows 2000 computer is responsible for a number of operating-system-specific functions. It must understand enough of the file system in use to find **Ntldr**, which is the program that locates and loads the Windows 2000 operating system files in the root folder. On a hard drive with a FAT partition, the partition boot sector is generally one sector long, and points to another location on disk that will ultimately permit the computer to find and launch Ntldr. On an NTFS partition, because the partition boot sector can be as many as 16 sectors long, it can contain all the necessary file system code needed to locate and launch Ntldr, without requiring transfer of control to another area on disk. Thus, the partition boot sector is responsible for loading a boot loader (Ntldr) into memory and initiating boot loader execution.

At this point, the **system partition**, the partition that contains the MBR and partition boot sector, must be on the first physical hard drive in the system. However, the **boot partition**—the partition that contains the Windows 2000 files—can be on the same partition, a different partition on the same drive, or on another drive entirely within the local computer. In other words, you boot Windows 2000 from the system partition, and run the operating system from the boot partition.

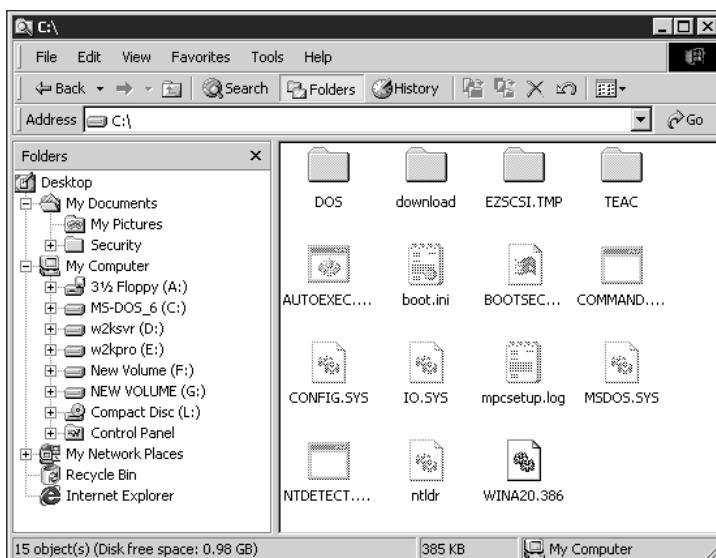


Because this terminology is counterintuitive (that is, it's the opposite of what you might expect) and because it appears on numerous Windows 2000-related Microsoft exams, it's important to remember this reversal of terminology.

## Boot Loader

Boot loader processing and files select an operating system to boot, and load the related operating system files from the boot partition. On PCs, once the boot OS is selected from the Boot.ini menu, Ntldr controls the operating system selection and hardware detection processes before the Windows 2000 kernel is initialized.

Ntldr, Boot.ini, Bootsect.dos, Ntdetect.com, and Ntbootdd.sys may all be present in the root directory of the startup disk (also known as the system partition; see Figure 14-3). Some files may be dimmed because they have the read-only attribute. The partition hosting the boot loader can be formatted with either FAT, FAT32, or NTFS. Of this collection of files, Ntldr, Ntdetect.com, and Boot.ini must always be present for Windows 2000 to boot (the other two are optional, and depend on the configuration of the particular machine in use).



**Figure 14-3** The system partition on a typical Windows 2000 system



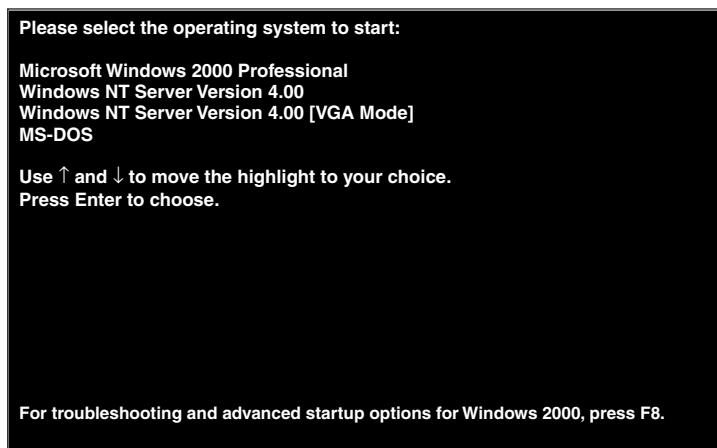
Bootsect.dos appears only if the machine has been configured to dual-boot between Windows 2000/Windows NT and DOS, Windows 3.x, or Windows 95. Ntbootdd.sys appears only when a SCSI controller has its built-in BIOS controller disabled; this file supplies the necessary controller driver that the hardware would otherwise provide.

At this point, Ntldr switches the processor into 32-bit flat memory mode. When an x86 computer starts, it is running in real mode, which means it is functioning as an old-fashioned 8088 or 8086 computer. Because Ntldr is a 32-bit program, it must change the processing mode to support the 32-bit flat memory model it uses before it can perform any further processing.

Next, Ntldr starts the appropriate file system. The code to access both FAT and NTFS file systems is programmed into Ntldr so that it can read, access, and copy files on either type of file system.

Next, Ntldr reads the Boot.ini file and displays the operating system selections it contains. The screen that appears at this point is usually called the boot loader screen or the **boot selection menu**, and represents the point at which users may select which operating system they would like to load (or which form of Windows 2000 graphics operation they would like to use).

A typical boot selection menu appears in Figure 14-4. Notice the prompt to access troubleshooting and advanced startup options by pressing F8. These options are discussed in a later section of this chapter.



**Figure 14-4** A typical Windows 2000 boot selection menu

14

Notice on this particular system, Windows 2000 Professional is present with Windows NT Server 4.0 and MS-DOS. In fact, Windows 2000 will coexist with numerous other operating systems, including those that depend on DOS for their underpinnings.

When you do not manually alter the highlighted selection of the boot menu, a line below the menu displays a counter: "Seconds until highlighted choice will be started automatically: 30". If a selection is not made before the counter reaches zero, the highlighted operating system will start automatically. To change the default operating system to load or the amount of time to wait before automatically loading the highlighted operating system, change the settings in the Boot.ini file, which will be discussed in greater detail later in this chapter.

If the user selects an operating system other than Windows 2000 or Windows NT, the boot loader loads Bootsect.dos and hands over control of the system. The other operating system then starts normally because Bootsect.dos contains the partition boot sector for that operating system. However, if the user selects a version of Windows 2000, the boot loader executes Ntdetect.com to gather hardware information.

The remaining functions of Ntldr (operating system selection, hardware detection, and configuration selection) are discussed later in this section. For now, note that Ntldr maintains control of the computer until it loads Ntoskrnl.exe and passes the hardware information and system control to that program.

## Detecting Hardware

Ntdetect.com is executed by the boot loader and is used to collect a list of hardware currently installed in the computer. Ntdetect checks the computer ID, bus/adapter type, video, keyboard, communication ports, parallel ports, floppy disks, and mouse or pointing devices. It creates a system profile that will be compared to Windows 2000 Registry entries that describe the system later during the boot process, at which point the operating system will look for discrepancies or potential problems.

Once hardware is detected, the system needs to select a system configuration, otherwise known as a hardware profile. If only a single hardware profile is defined, this will always be used. If two or more hardware profiles are present, the system will attempt to select a profile based on detected hardware. If the system cannot make an automatic selection, you'll be prompted to manually select a hardware profile.

---

## TROUBLESHOOTING AND ADVANCED STARTUP OPTIONS

Windows 2000 has combined the boot and recovery options of Windows NT and Windows 95/98. The result is a more robust operating system and additional options to restore a malfunctioning system to a functional state. To access the additional startup options, when the boot menu appears, press F8 before the timer expires. Once F8 is pressed, the Advanced Options Menu appears (see Figure 14-5).

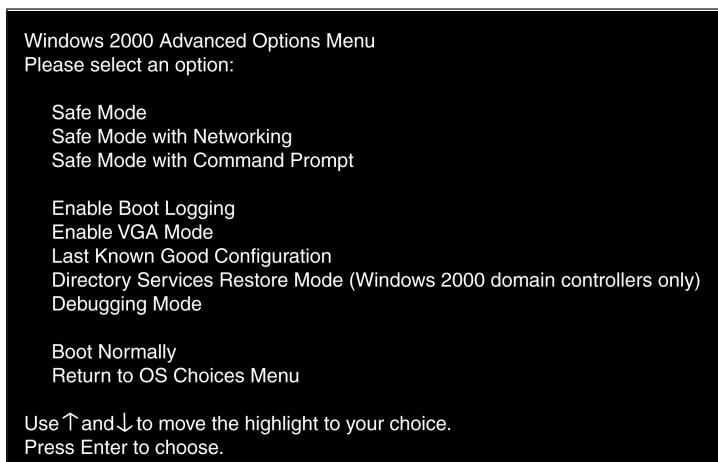


Figure 14-5 The Advanced Options Menu

The contents of this menu are somewhat dependent on installed components, such as the Remote Installation Service, but it typically contains the following items:

- *Safe Mode*—Boots Windows 2000 with only the minimum required system files and device drivers. Safe Mode does not load networking components (see Hands-on Project 14-3).
- *Safe Mode with Networking*—Boots Windows 2000 in the same manner as Safe Mode, but adds networking components (see Hands-on Project 14-6).
- *Safe Mode with Command Prompt*—Boots Windows 2000 in the same manner as Safe Mode, but boots to a command prompt instead of to the GUI environment.
- *Enable Boot Logging*—A toggle that enables or disables the boot process and writes details to a log file regarding drivers and services. The log file is located at %systemroot%\Ntbtlog.txt.
- *Enable VGA Mode*—Boots Windows 2000 normally, but uses only the basic VGA video driver (see Hands-on Project 14-5).
- *Last Known Good Configuration*—Boots Windows 2000 with the **Last Known Good Configuration (LKGc)**, the state of the Registry as it was recorded during the last successful user logon (see Hands-on Project 14-4).
- *Directory Services Restore Mode*—Valid only on Windows 2000 domain controllers; used to boot Windows 2000 and restore Active Directory.
- *Debugging Mode*—Boots Windows 2000 normally, but sends debugging information to another system over a serial cable. Details about using this option are included in the *Windows 2000 Resource Kit*.

Advanced Options for booting can be used to recover from a wide variety of system problems or failures. Safe Mode offers the ability to boot into a functioning system even when specific drivers are corrupted or failing. This includes bypassing bad video drivers, network drivers, and even GUI controls by booting into Enable VGA Mode, Safe Mode (without networking support), and/or Safe Mode with Command Prompt, respectively. In most cases, this allows you to replace or remove the problematic driver before rebooting back into normal mode.



If a problem is occurring and you are unable to discern its exact cause or nature, you may want to choose Enable Boot Logging from the Advanced Options menu to record the process of steps performed between the boot menu and the logon prompt. The resultant file, %systemroot%\Ntbtlog.txt, may provide clues as to the driver, system, or procedure that is causing the system malfunction.

If you've recently installed a driver or even an entire software product, or just modified the Registry, and the result is a system that will not fully boot, the Last Known Good Configuration is a great first step in returning the system to a functional state. The LKGC will return the system to the state of the Registry at the time of the last successful logon.

If none of these options provides you a method to restore your system, you do have one final option from the Advanced Options Menu, namely Debugging Mode. This mode is used in conjunction with a second computer connected by a serial cable. Debugging Mode causes the boot process to send detailed information on activities to the companion system. This information can be used to determine at what point in the boot process problems are occurring. The information created by Debugging Mode is rather complex and is typically used only by high-end programmers. If you want more details on the Debugging Mode process and how to interpret the extracted data, please consult the *Windows 2000 Resource Kit*.

---

## BOOT CONFIGURATION AND SELECTING AN OPERATING SYSTEM

The Windows 2000 boot configuration can be controlled through its configuration file, Boot.ini. As mentioned earlier, Boot.ini is located in the root directory of the system partition, and is used by the boot loader to display the list of available operating systems. This file consists of two sections: [boot loader] and [operating systems]. A typical Boot.ini file is shown in Figure 14-6.

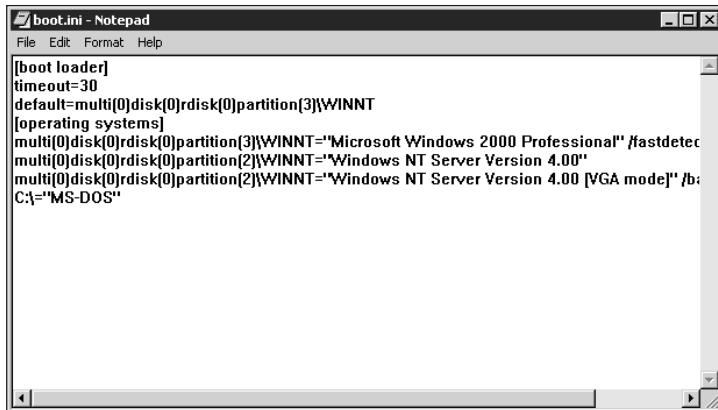


Figure 14-6 Boot.ini viewed through Notepad

### [boot loader]

The [boot loader] section of the Boot.ini file contains two items: timeout and default. The *timeout* setting defines the number of seconds the system will wait for the user to select an operating system before loading the default operating system. If *timeout* is set to zero, Ntldr immediately loads the default operating system without displaying the boot loader screen. To cause the system to wait indefinitely for a selection, set the *timeout* to -1. This setting, however, can only be altered by using a text editor, because it is an illegal value for the setting from the System icon in the Control Panel. (See the section later in this chapter on editing Boot.ini, which explains how to edit this file, and what kind of text editor to use.) The *default* setting in Boot.ini lists the path to the default operating system.

## [operating systems]

The [operating systems] section of Boot.ini lists the available operating systems. Each listing contains the path to the boot partition for the operating system, the text displayed in the boot loader screen, and optional parameters. The text is clipped in the screen capture in Figure 14-6, but here's what it looks like in its entirety:

```
multi(0)disk(0)rdisk(0)partition(3)\WINNT="Microsoft Windows
    2000 Professional" /fastdetect
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Windows NT Server
    Version 4.00"
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Windows NT Server
    Version 4.00 [VGA mode]" /basevideo /sos
C:\="MS-DOS"
```

The following list details some of the switches that can be added to the end of entries in the [operating systems] section of Boot.ini. In most cases, you'll want to employ the F8 Advanced Options menu (see earlier this chapter) to access troubleshooting boot methods. However, you can employ the following switches and switch combinations to mimic the Advanced Options menu selections in your Boot.ini file:

- **/BASEVIDEO**—Starts Windows 2000 in standard VGA mode (640 × 480)
- **/DEBUG**—Loads the debugger and allows access by a host debugger connected to the computer
- **/SOS**—Displays the device driver names when they are loaded
- **/SAFEBOOT:MINIMAL**—Boots into Safe Mode
- **/SAFEBOOT:NETWORK**—Boots into Safe Mode with Networking
- **/SAFEBOOT:MINIMAL(ALTERNATESHELL)**—Boots into Safe Mode with Command Prompt
- **/SAFEBOOT:DSREPAIR**—Boots into Directory Services Restore Mode (domain controllers only)
- **/BOOTLOG**—Enables boot logging
- **/NOGUIBOOT**—Boots without showing the splash screen. Does not determine whether Windows 2000 GUI environment or command prompt is booted
- **/FASTDETECT**—Performs a verification of hardware components rather than a full inspection, to speed boot time



The switches used in the Boot.ini file are not case sensitive.

## Advanced RISC Computing (ARC) Pathnames

In the Boot.ini file, the path pointing to the \winnt directory is written using the **Advanced RISC Computing (ARC) pathname** naming conventions. These pathnames are described as follows:

- *scsi(n) or multi(n)*: This portion of the path indicates the type of the device on which the operating system resides. *scsi* is used if the operating system is on a SCSI hard disk that is connected to a SCSI adapter that has a disabled built-in BIOS. *multi* is used for other hard disks including IDE, EIDE, and SCSI with a built-in BIOS. The *(n)* indicates the hardware adapter from which to boot. It is replaced with a number corresponding to the correct hardware adapter, numbered ordinally (starting with zero).
- *disk(n)*: This portion of the path indicates which SCSI bus number should be used. The *(n)* always equals zero when the adapter is a multiadapter (that is, the ARC494 path starts with *multi*). Otherwise, it is numbered ordinally.
- *rdisk(n)*: This portion of the path indicates the SCSI LUN number or selects which of the hard disks attached to the adapter contains the operating system. *(n)* always equals zero when the adapter is SCSI. Otherwise, it is numbered ordinally.
- *partition(n)*: This portion of the path selects the disk partition that contains the operating system files. Partition is numbered cardinally (starting with 1).
- *\path*: The final portion of the path indicates the directory on the partition in which the operating system files are found. The default path for Windows 2000 is \Winnt.

---

## EDITING BOOT.INI

To make changes to a Boot.ini file, the user has two options: to use the Control Panel to edit this file indirectly, or to use a text editor to change the file directly.

### Using the Control Panel

Using the Control Panel to make changes to Boot.ini is the safest way to proceed (try Hands-on Project 14-1). By opening the System applet in the Control Panel, selecting the Advanced tab, then clicking the Startup and Recovery button (see Figure 14-7), you can make certain changes to your setup. The Startup and Recovery dialog box (shown in Figure 14-8) allows you to choose a default boot selection and to select a delay interval before the boot selection starts automatically. This delay time corresponds to the timeout value set in Boot.ini. These options are depicted in the System startup pane in the Startup and Recovery dialog box. Notice also that the options that control debugging output for system failures appear on this pane as well; this information often comes in handy when severe problems manifest.

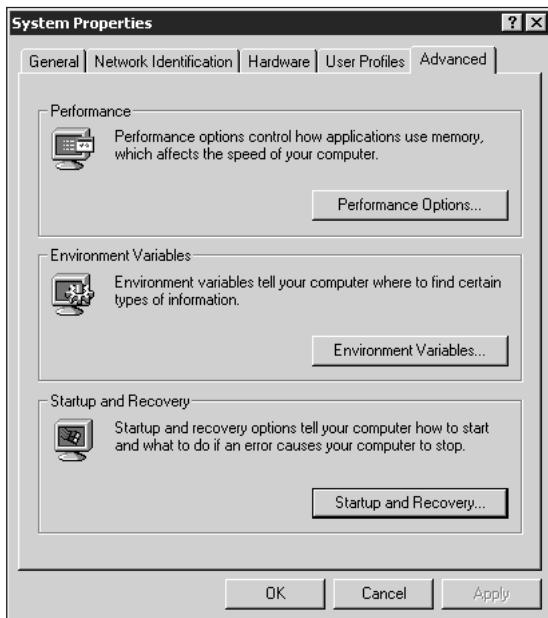


Figure 14-7 The Advanced tab of the System Control Panel applet

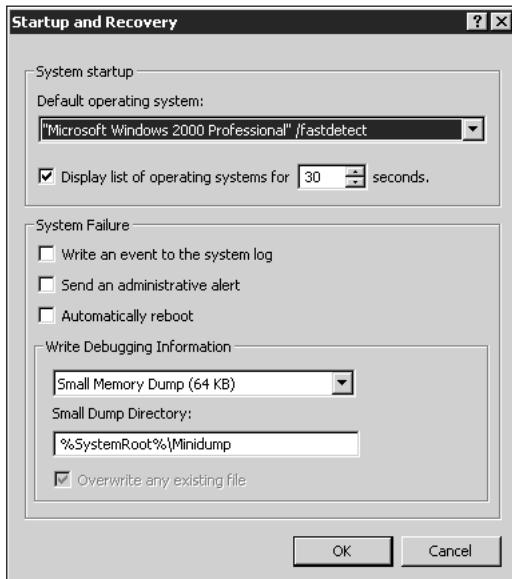


Figure 14-8 Startup and Recovery dialog box

## Using a Text Editor

You can use Notepad or any other text editor to edit Boot.ini (try Hands-on Project 14-2). As with any initialization file, you should be careful when editing the file. If you configure the file incorrectly, Windows 2000 may not boot. You should always create a backup copy of the file and name it Boot.bak before you make any changes.



If you do decide to edit the Boot.ini file from a text editor, you must first reset the file's attributes to make it editable. By default, this file has the following attributes set: System, Hidden, and Read-only. To edit the file, you must reset it to turn off at least the Read-only attribute (we usually turn off the Hidden attribute as well). To change the file's attributes, locate the file in Windows Explorer, right-click its entry, click Properties, and then uncheck the attribute settings boxes on the General tab that show up by default (see Figure 14-9). After you make these attribute changes, you can edit the file using any simple text editor. (Notepad works fine.)

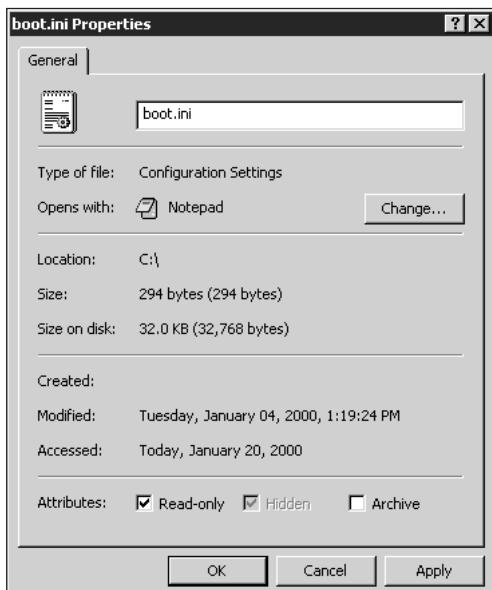


Figure 14-9 The Properties dialog box of the Boot.ini file

## WINDOWS 2000 LOAD PHASE

The Windows 2000 load phase begins when the kernel assumes control of the machine. It consists of the following five stages:

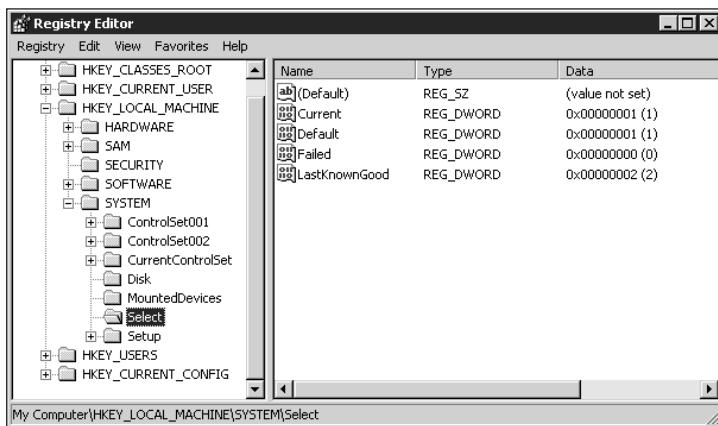
- Loading the kernel
- Initializing the kernel

- Services load
- Windows 2000 system start
- Logging on

## Loading the Kernel

Once you've selected the option to boot into Windows 2000, a brief "Starting Windows..." text message is displayed before the full Windows 2000 splash screen is shown. This image includes a loading status thermometer bar that gives you a minor indication of progress toward loading. While you are "entertained" by the slow progression of the blue squares, the boot loader loads the Windows 2000 kernel (Ntoskrnl.exe) and the hardware abstraction layer (HAL; file Hal.dll) into memory. However, these programs are not executed at this time. Before executing the programs, the boot loader loads the Registry key HKEY\_LOCAL\_MACHINE\SYSTEM from the %systemroot%\System32\Config\System directory.

At this point, the boot loader retrieves the configuration you selected from the Registry subkey HKEY\_LOCAL\_MACHINE\SYSTEM\Select. Based on the ControlSet00x setting in the subkey, the boot loader knows which ControlSet00x to use. For example, if you chose the Last Known Good Configuration option during the configuration selection process, the **control set** may be ControlSet003 rather than ControlSet001, which is the default configuration. Notice the values of the Current, Default, Failed, and LastKnownGood value entries in Figure 14-10. (Working with the Registry and the contents of Registry keys are covered in Chapter 13.)



**Figure 14-10** The HKEY\_LOCAL\_MACHINE\SYSTEM\Select subkey

The boot loader then loads the drivers listed in the Registry subkey HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services. These drivers are loaded and/or initialized according to their settings in the Registry.

## Initializing the Kernel

After its initialization, the kernel creates the Registry key HKEY\_LOCAL\_MACHINE\HARDWARE, using the information received from the boot loader. This key contains the hardware information that is computed when the system is started up, and includes information about components on the system board and the interrupts used by specific hardware devices.

The kernel also creates the CloneControlSet by making a copy of the CurrentControlSet. The Clone set is never modified, because it is intended to be an identical copy of the data used to configure the computer and should not be modified during the startup process.

The kernel then initializes the drivers that were loaded by the boot loader. If drivers experience errors as they load, they send conditions to the kernel that determine how the error is treated. The error levels are as follows:

- *Ignore*: The error is ignored and no message is displayed to the user, if the ignore condition is sent to the kernel.
- *Normal*: The boot process continues, but a message is displayed to the user if the device driver returns the normal error condition.
- *Severe*: The management of this error depends on whether the Last Known Good Configuration is in use or not. If the LKGC is not being used, then the error is displayed to the user, and the boot process restarts using the LKGC. If the LKGC is already in use, then the message is displayed and the boot process continues.
- *Critical*: The management of this error depends on whether the LKGC is in use or not. If not, then the error is displayed to the user, and the boot process restarts using the Last Known Good Configuration. If the LKGC is already in use, then the message is displayed and the boot process fails.

All such events are saved automatically in the System log, and invoke on-screen messages as well. The System log is available as one of the views in the Windows 2000 Event Viewer (Start, Settings, Control Panel, Administrative Tools, Event Viewer), and should always be checked whenever errors are reported during the boot process. Because that process cannot be interrupted, however, it's necessary to wait and inspect the log after the bootup phase is complete.

## Services Load

During the services load phase, the kernel starts the Session Manager. The Session Manager reads the entries that are stored in the Registry key: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager.

It then starts programs that correspond to the key entries under this Registry key: HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\BootExecute.

The default entry for this key is “autocheck autochk \*”. Autocheck makes sure that the files stored on your hard drive are always consistent. It detects and attempts to repair damaged files and directories. As with any repair utility, it cannot guarantee that all files can be fixed or retrieved.

Once Autocheck is complete, the paging files are set up. These are stored under HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management.

The Session Manager then writes the CurrentControlSet and the CloneControlSet to the Registry, and, finally, loads the subsystems that are defined in the Registry.

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Subsystems contains the subsystem information. The Windows (Win32) subsystem is the default subsystem for Windows 2000, and is also the subsystem within which the default user shell always executes.

## Windows 2000 System Startup

Once the Windows 2000 services have all started, and the elements in the group of processes that are configured to launch on startup are fired off, the Windows 2000 system may be considered to be fully started. This brief, but meaningful, phase of the process is signaled by the appearance of the Windows 2000 logon screen as the Win32 subsystem starts WinLogon.exe, and that process automatically launches the Local Security Administration (Lsass.exe) process.

## Logging On

Unlike Windows NT Workstation 4.0, Windows 2000 Professional does not require the issuing of the Ctrl+Alt+Delete keystroke to initiate the logon dialog box by default. The “Disable Ctrl+Alt+Delete requirement for logon” option in the Security Options section of the Local Security Policy is enabled by default. If you alter this setting, you’ll need to issue the keystroke to access the logon dialog box.

Until a user successfully logs on, the boot process is not complete. Once a user logs on, the Clone control set is copied to the Last Known Good control set. This procedure provides the values that will be used the next time the machine is powered up, if the user elects to invoke the Last Known Good Configuration.

14

---

## MULTIPLE-BOOT SYSTEMS

One of the biggest advantages of the Windows 2000 operating system is its ability to peacefully coexist with other operating systems. Each operating system uses one or more file systems to organize the data within the volumes. Some operating systems can use the same file system, whereas others are incompatible. For example, MS-DOS, Windows 95/98, Window NT, and Windows 2000 are able to share files via FAT volumes, and Windows NT and Windows 2000 are able to share files via NTFS volumes. Windows 2000 and UNIX do not have a common file system, although it is possible for Linux to access FAT volumes.

## Multiple Windows Operating Systems

Windows 3.1, Windows 3.11, Windows 95/98, and Windows NT can all exist on the same system as Windows 2000. When Windows 2000 is to be installed on a system with another operating system—especially some previous version of Windows—it is important to specify a different partition for installation. Unless you want to upgrade the computer (that is, install Windows 2000 over an existing OS), always define a new main directory and partition, different from the one already in use. Different versions of Windows 2000 may also be installed on the same computer, but, again, each must have a separate partition.

If you plan to use applications from the different versions of Windows you have installed, you must install the application from each operating system. For example, if you intend to use Microsoft Word from both Windows 98 and Windows 2000, you must run the Word setup program while the computer is booted to each operating system.

## Multiple Installation Order

When installing multiple operating systems on x86-based computers, the order in which you install the OSs is important. When installing Windows 2000 and MS-DOS, it is best to install MS-DOS first, then Windows 2000. Windows 2000 will see the DOS operating system and leave it intact. The same guideline applies to installing Windows 2000 and Windows 95/98 or Windows NT. At this time, the recommended installation order is: older Windows products first, then Windows 2000. If you plan on running all three operating systems (MS-DOS, Windows 98, and Windows 2000), they should be installed in that order: MS-DOS, then Windows 98, and Windows 2000 last. When installing multiple versions of Windows 2000 or Windows NT onto the same system, it really doesn't matter which one is installed first. As a general rule, install the newest operating system last and the oldest first.

---

## CHAPTER SUMMARY

- ❑ The Windows 2000 boot process can be daunting, but it is not nearly as mysterious as one first supposes. It follows the same general boot steps as any other operating system, and in fact, “plays well with others.” After the POST (power-on self test), the BIOS loads the Master Boot Record (MBR), which then loads the partition boot sector. Then the boot loader takes control of the system and begins the true Windows 2000 boot. The user is presented with options for choosing the operating system to load, and—if he or she chooses Windows 2000, the configuration to use.
- ❑ When the boot menu appears, you may press F8 to access Advanced Options. The Advanced Options are alternate boot methods that can bypass certain types of drivers or subsystems to aid in troubleshooting. Advanced Options include, among others, Safe Mode, Enable VGA, Enable Boot Logging, and Last Known Good Configuration.
- ❑ After the boot loader, the kernel is loaded into memory and is granted control of the computer. The kernel loads the operating system files and device drivers before finally allowing the user to log on. When the user successfully logs on to the computer, it is considered a good startup and the configuration is saved to the Registry.

- The boot process can be altered by changing the Boot.ini file. This includes information such as the default operating system, its location, and the amount of time to wait before automatically loading the default OS. The type of information displayed and the debugger setting can be changed by adding switches to the configurations in the Boot.ini file.
- Windows 3.1, Windows 3.11, Windows 95/98, and Windows NT can all exist on the same system as Windows 2000. You can configure Windows 2000 to offer the choice of booting to other operating systems. In a multiboot system, it's generally best to install in order from older operating systems to newer.

---

## KEY TERMS

**Advanced RISC Computing (ARC) pathname** — Naming convention used in the Boot.ini file to define the particular hard disk and partition where Windows 2000 operating system files reside.

**BIOS (basic input/output system)** — A special PC ROM chip that contains sufficient program code to let a computer perform a POST routine, to check its hardware components, and to operate basic input and output routines for keyboard or mouse input, and screen output.

**boot partition** — In Windows 2000, the disk that contains the Windows 2000 operating system files.

**boot phase** — Any of a number of stages in the Windows 2000 boot process, starting with the power-on self test (POST), through initial startup activities, to activation of a boot loader program, to selection of the operating system (or version) to boot, to hardware detection (Ntdetect), to selecting a configuration.

**boot process** — The process of bringing up a completely functional computer, starting from initial power-up (or reboot) through the boot phases and load phases involved in starting the hardware, finding a boot loader, and then loading and initializing an operating system.

**boot selection menu** — The list of bootable operating systems (or versions) that Boot.ini provides for display at the end of the Windows 2000 boot phase.

**CMOS (complementary metal-oxide semiconductor)** — A special, battery-powered chip that can store not only the software necessary to conduct the POST, but also the basic, nonvolatile configuration information that POST uses to check the RAM installed in a system, the number and type of hard drives, the type of keyboard and mouse, and so forth.

**control set** — A special set of Registry values that describes a Windows 2000 machine's startup configuration that is saved each time a Windows machine is shut down (as the current configuration) and each time a user successfully logs on for the first time after bootup (as the Last Known Good Configuration).

**Last Known Good Configuration (LKGc)** — The control set for Windows 2000 that is automatically saved by the system in a special set of Registry keys the first time a user logs on successfully to a system immediately after it has booted up. This information provides a safe fallback to use when booting the system the next time, if changes made to the Registry in the interim cause problems with booting (or if changes have been introduced that a user does not wish to retain on that system).

**load phase** — The Windows 2000 load phase begins when the kernel assumes control of the machine, and consists of the following five steps: (1) loading the kernel, (2) initializing the kernel, (3) loading services, (4) starting the Windows 2000 system, and (5) logging on. All five steps must be completed successfully for a complete load to occur.

**Master Boot Record (MBR)** — The partition table for a disk, and the code that permits that partition table to be read. A functioning MBR is required to boot a hard disk.

**Ntldr** — The Windows 2000 loader program that manages the boot and load phases of Windows 2000 on a PC.

**partition boot sector** — The partition that contains the information the file system uses to access the volume, including a physical description of the disk, the name and version of the operating system files, the bootstrap code, and an instruction that allows the Master Boot Record to find all this information.

**power-on self test (POST)** — The system check performed by all computers when they are turned on.

**system partition** — In Windows 2000, the disk that contains the MBR and partition boot sector.

---

## REVIEW QUESTIONS

1. The \_\_\_\_\_ contains the files that load the initial components of the operating system.
  - a. boot partition
  - b. system partition
  - c. start partition
  - d. kernel partition
2. What program has control of an x86 computer when the user is able to choose which operating system to boot?
  - a. Ntldr
  - b. Osloader
  - c. Boot.ini
  - d. Ntbootdd.sys
3. When configuring an x86 computer for multiple operating systems, \_\_\_\_\_ should always be loaded last.
4. When booting an x86 computer, the boot loader must be installed on a(n) \_\_\_\_\_ file system. (Choose all that apply.)
  - a. NTFS
  - b. HPFS
  - c. FAT
  - d. FAT32

5. The \_\_\_\_\_ setting in the Boot.ini file defines the operating system that will be automatically loaded.

- [operating systems]
- [system loader]
- [boot loader]
- default

6. By default, Windows 2000 Professional requires you to press the Ctrl+Alt+Delete key-stroke sequence to access the logon dialog box. True or False?

7. \_\_\_\_\_ is the primary boot loader for x86-based systems.

- Ntldr
- Osloader.exe
- Bootdd.sys
- Bootsect.dos

8. The timeout option is in the \_\_\_\_\_ section of the Boot.ini file.

9. What portion of a computer system startup is the same on all computers?

- POST
- initial startup
- boot loader
- OS selection

10. The Boot.ini file can be changed by one of two methods. What are they?

- Control Panel System applet
- Control Panel Startup applet
- Windows 2000 Configuration Manager
- using a plaintext editor

11. The \_\_\_\_\_ file is only accessed when SCSI disks with onboard BIOS disabled are used.

- Multidisk.sys
- Scsildr.sys
- Rdisk.sys
- Ntbootdd.sys

12. Which of the following are selections listed on the Advanced Options menu when you press F8 during the boot menu display? (Choose all that apply.)

- Safe Mode with Command Prompt
- Enable VGA Mode
- NTFS Transfer Mode
- Debugging Mode

13. Which ARC settings are used only for SCSI controllers without an enabled onboard BIOS? (Choose all that apply.)
  - a. multi()
  - b. scsi()
  - c. rdisk()
  - d. disk()
14. The Ntoskrnl.exe file is located in the \_\_\_\_\_ directory on an x86 system.
15. The Last Known Good Configuration is accessed by pressing the spacebar after the operating system is selected from the boot menu. True or False?
16. You recently installed a new video driver and a new networking interface driver, and now your system will not boot, or at least you never see the logon prompt. Which of the following advanced options should you use to attempt to return to a fully functional system?
  - a. Safe Mode
  - b. Safe Mode with Networking
  - c. Safe Mode with Command Prompt
  - d. Enable VGA Mode
  - e. Enable Debugging Mode
  - f. Enable Boot Logging
17. The Boot.ini parameter switches that mimic the Safe Mode with Networking Advanced Options menu selection are \_\_\_\_\_?
  - a. /SAFEBOOT:MINIMAL(ALTERNATESHELL) /SOS /BOOTLOG /NOGUIBOOT
  - b. /SAFEBOOT:DSREPAIR /SOS
  - c. /SAFEBOOT:NETWORK /SOS /BOOTLOG /NOGUIBOOT
  - d. /BOOTLOG
18. The \_\_\_\_\_ Boot.ini switch displays the names of the device drivers as they are loaded.
  - a. /B
  - b. /AT
  - c. /SOS
  - d. /DRV
19. The Last Known Good Registry key is written or updated after \_\_\_\_\_.
20. The \_\_\_\_\_ partition contains the Windows 2000 operating system files.
  - a. system
  - b. boot
  - c. start
  - d. kernel

21. If the system and boot partitions are the same and reside on an IDE hard drive, which of the following ARC names would appear in Boot.ini?
  - a. scsi(0)disk(0)rdisk(1)partition(1)
  - b. multi(0)disk(0)rdisk(0)partition(1)
  - c. multi(0)disk(1)rdisk(0)partition(1)
  - d. multi(0)disk(0)rdisk(1)partition(1)
22. Windows 2000 uses its own built-in input/output logic and drivers, and ignores whatever BIOS is installed in a computer. True or False?
23. The presence of a floppy in drive A can cause which of the following situations? (Choose all that apply.)
  - a. booting from the floppy to whatever OS is installed there
  - b. failure to boot due to a missing boot sector on the floppy
  - c. “Non-system disk or disk error: Replace and press any key when ready” error message
  - d. normal booting from the hard drive
24. The same MBR is found on Windows 95, MS-DOS, Windows NT, Windows 2000, and Windows 3.x systems. True or False?
25. Ntbootdd.sys appears on a system when which of the following operating systems is present in a multiboot configuration with Windows 2000?
  - a. MS-DOS
  - b. Windows 3.x
  - c. Windows 98
  - d. none of the above

---

## HANDS-ON PROJECTS



### Project 14-1

#### To modify the Boot.ini file by using Control Panel:

1. Open the Control Panel (**Start, Settings, Control Panel**).
2. Double-click the **System** applet to start it.
3. Select the **Advanced** tab (refer to Figure 14-7).
4. Click the **Startup and Recovery** button. This reveals the Startup and Recovery dialog box (refer to Figure 14-8).
5. Notice that the **Startup** option defines the default operating system for the computer. Select another operating system from the drop-down list.

6. To modify the amount of time the list appears when the system is booted, change the **Display list of operating systems for** option. Change this setting to **10** seconds by clicking on the down arrow beside the field.
7. The remaining options define what action the kernel will take when a STOP error occurs. In most situations, these should not be changed.
8. To save the configuration to Boot.ini, click **OK**.
9. To see the effect of the changes you made, restart the computer.
10. Repeat this process to restore the original settings.



## Project 14-2

### To change the Boot.ini settings using a text editor:

1. First create a backup copy of the Boot.ini file: Launch Windows Explorer (**Start, Programs, Accessories, Windows Explorer**); select the root of drive C in the left pane. Right-click drive **c:**, select Folder Options, and make sure that the options for viewing file extensions, hidden files, and protected operating system files are checked. Right-click the **Boot.ini** entry, select the **Copy** entry, then click on the current drive and press **Ctrl+V** (paste). This creates a file named “Copy of boot.ini” in that directory. Rename the file to **Boot.bak**. You may need this if something goes wrong later.
2. Next, the Read-only attribute may have to be removed from the Boot.ini file. To do this, highlight the **Boot.ini** file.
3. Right-click the file and select **Properties**.
4. If the box next to **Read-only** is checked, deselect it. This changes the file’s attributes so that it can be altered (refer to Figure 14-9).
5. Click **OK**.
6. Open the Boot.ini file in Notepad: open the **Start** menu, select **Run**, enter **notepad c:\boot.ini** in the box provided, and click **OK**. Notepad opens with the Boot.ini file displayed (refer to Figure 14-6).
7. Restore the timeout to 30 seconds by changing the **timeout=** value to **30**.
8. Save the file by selecting **File, Save**.
9. Exit Notepad by selecting **File, Exit**.
10. Reboot the computer to deploy your changes.



## Project 14-3

### Rebooting Windows 2000 into Safe Mode:

1. Enter the Windows 2000 attention sequence (**Ctrl+Alt+Delete**) to invoke the Windows 2000 Security dialog box. Click the **Shut Down** button, select **Restart** from the pull-down list, and then click **OK**.
2. As Windows 2000 reboots, watch for the boot selection menu. As soon as it appears, press the **F8** key. This reveals the Advanced Options menu. (Refer to Figure 14-5.)
3. Use the arrow keys on the keyboard to select **Safe Mode** from the list of options.

4. Press the **Enter** key, and allow the boot process to continue to completion.
5. Your system will boot with minimal drivers and without network support.
6. Repeat step 1, and allow your machine to reboot normally.



## Project 14-4

### Rebooting Windows 2000 with the Last Known Good Configuration:



Performing this project will cause all changes made to the system since the last successful logon to be discarded.

1. Enter the Windows 2000 attention sequence (**Ctrl+Alt+Delete**) to invoke the Windows 2000 Security dialog box. Click the **Shut Down** button, select **Restart** from the pull-down list, and then click **OK**.
2. As Windows 2000 reboots, watch for the boot selection menu. As soon as it appears, press the **F8** key. This reveals the Advanced Options menu (refer to Figure 14-5).
3. Use the arrow keys on the keyboard to select **Last Known Good Configuration** from the list of options.
4. Press the **Enter** key, and allow the boot process to continue to completion.
5. Your system will boot with the state of the Registry recorded at the last successful logon.



## Project 14-5

### Rebooting Windows 2000 with minimal VGA support:

14



This boot method should be used when a bad video driver is present or incorrect resolution has been set.

1. Enter the Windows 2000 attention sequence (**Ctrl+Alt+Delete**) to invoke the Windows 2000 Security dialog box. Click the **Shut Down** button, select **Restart** from the pull-down list, and then click **OK**.
2. As Windows 2000 reboots, watch for the boot selection menu. As soon as it appears, press the **F8** key. This reveals the Advanced Options menu (refer to Figure 14-5).
3. Use the arrow keys on the keyboard to select **Enable VGA Mode** from the list of options.
4. Press **Enter**, and allow the boot process to continue to completion.
5. Your system will boot normally, but will use the standard VGA video drivers at  $640 \times 480$  with a color depth of 16 or 256 (depending on your video card). This will allow you to correct your display resolution or replace the bad video driver.



## Project 14-6

### Rebooting Windows 2000 into Safe Mode with Networking:

1. Enter the Windows 2000 attention sequence (**Ctrl+Alt+Delete**) to invoke the Windows 2000 Security dialog box. Click the **Shut Down** button, select **Restart** from the pull-down list, and then click **OK**.
2. As Windows 2000 reboots, watch for the boot selection menu. As soon as it appears, press the **F8** key. This reveals the Advanced Options menu (refer to Figure 14-5).
3. Use the arrow keys on the keyboard to select **Safe Mode with Networking** from the list of options.
4. Press **Enter**, and allow the boot process to continue to completion.
5. Your system will boot with minimal drivers but will include network support. This boot method is useful when attempting to troubleshoot a system that requires network access to tools, data files, or traffic.

---

## CASE PROJECTS



1. The Engineering Department in your company has decided to update their computers to Windows 2000 Professional. They currently have four PCs, two running Windows 3.11 and two running Windows 95. They would like to retain their current configurations and programs. Outline the steps necessary to install Windows 2000 on their systems and explain what configurations will be available after the update is complete.
2. After installing a new graphics controller on a Windows 2000 Professional machine, you start up the system, but when the boot process is complete, you see nothing on the monitor except a small dot of light in the exact center. What boot option can you use to see enough of the screen to try a different driver, or to change display settings in the Display Control Panel applet?
3. By default, the Boot.ini entry for Windows 95 in the boot selection menu reads “MS Windows.” How might you edit Boot.ini to change this value to read “Windows 95 Rules!” instead? What part of the Boot.ini file does the appropriate entry reside in, and which entry should you edit?